TITLE:
Proving Correctness of Algol-Like Programs in a Formal System (アルゴリズムにおける証明論)

AUTHOR(S):
ONO, HIROAKIRA

CITATION:
ONO, HIROAKIRA. Proving Correctness of Algol-Like Programs in a Formal System (アルゴリズムにおける証明論). 数理解析研究所講究録 1975, 236: 124-130

ISSUE DATE:
1975-05

URL:
http://hdl.handle.net/2433/105502

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

数理解析研究所講究録
236巻 1975年 124-130

Proving correctness of Algol-like programs in a formal system

Hiroakira ONO*

0. Introduction

In this paper, we shall introduce a formal system $\mathcal{S}$, in which we can prove the (partial) correctness of Algol-like programs. The method used to construct the system $\mathcal{S}$ is essentially based on Hoare [2]. But in our system we can give a proof of the correctness of programs in a completely formal manner. Our system is a version of that in [4]. We shall compare the system $\mathcal{S}$ with the inductive assertion method (see e.g. [3]), by using the infinitary language. We intend to construct $\mathcal{S}$ rather for its formal properties than for its practical usefulness.

1. Formal system

Before introducing $\mathcal{S}$, we shall define the class of programs, called Algol-like programs.

Definition 1. Statements are defined inductively as follows.

1) An expression $y:=f(x_1,\ldots,x_m)$ is a statement, where $x_1,\ldots,x_m$ and y are variables and f is an m-ary function symbol.

2) If S_1 and S_2 are statements and P is an n-ary predicate symbol, then <u>if</u> $P(x_1,\ldots,x_n)$ <u>then</u> S_1 <u>else</u> S_2 is a statement.

3) If S is a statement and P is an n-ary predicate symbol, then <u>while</u> $P(x_1,\ldots,x_n)$ <u>do</u> S is a statement.

4) If $S_1,\ldots, S_n$ ($n>0$) are statements, then <u>begin</u> S_1; S_2; ... ; S_n <u>end</u> is a statement.

---

* Tsuda College

Any statement of the form of 4) is called an Algol-like program.

Formulas of the system $\mathcal{S}$ are the same as those of the first order predicate calculus. We use $\neg$, $\wedge$, $\vee$, $\forall$, $\exists$ as logical connectives. A formula $A \supset B$ is considered as an abbreviation of the formula $\neg A \vee B$. $A_x[y]$ denotes the formula obtained from A by replacing each free occurrence of x in A by y. We assume that function symbols and predicate symbols appearing in the definition of statements are contained in the language of $\mathcal{S}$.

The system $\mathcal{S}$ is a Gentzen-type one. We use the letters Γ, Γ', Δ, Δ', Θ, Π etc. to denote finite sets of formulas. $\Gamma_x[t]$ denotes the set of formulas which is obtained from Γ by replacing each free occurrence of x in every formula in Γ by a term t. Now, let S be a statement or empty and Δ is a nonempty set. Then $\Gamma \xrightarrow{S} \Delta$ is a sequent of $\mathcal{S}$. Informally, this expression means that if every formula in Γ holds, then <u>every</u> formula in Δ holds after the execution of the statement S terminates. Thus the above sequent is equivalent to the expression $A_1 \wedge \dots \wedge A_m \{ S \} B_1 \wedge \dots \wedge B_n$ in Hoare [2], where $\Gamma = \{A_1, \dots, A_m\}$ and $\Delta = \{B_1, \dots, B_n\}$. In particular, when S is empty, the above sequent has the same meaning as the sequent $\Gamma \longrightarrow B_1 \wedge \dots \wedge B_n$ in Gentzen's LK [1]. Sometimes we write sets of formulas of the form $\Gamma \cup \{A\}$ and $\Gamma \cup \Gamma'$ as Γ, A and Γ, Γ', respectively.

Any sequent of the form $\Gamma \rightarrow \Gamma$ is a beginning sequent of $\mathcal{S}$. Rules of inference of $\mathcal{S}$ are as follows, where S is a statement or empty.

1) $$\frac{\Gamma \xrightarrow{S} \Delta}{\Gamma, \Gamma' \xrightarrow{S} \Delta}$$

2a) $$\frac{\Gamma \xrightarrow{S} \Delta \quad \Delta \longrightarrow H}{\Gamma \xrightarrow{S} H}$$

2b) $$\frac{\Gamma \longrightarrow \Delta \quad \Delta \xrightarrow{S} H}{\Gamma \xrightarrow{S} H}$$

3) $$\frac{\Gamma \longrightarrow A, \Delta}{\neg A, \Gamma \longrightarrow H}$$

4) $$\frac{A, \Gamma \longrightarrow B}{\Gamma \longrightarrow \neg A \vee B}$$

5a) $$\frac{A, \Gamma \xrightarrow{S} \Delta}{A \wedge B, \Gamma \xrightarrow{S} \Delta}$$

5b) $$\frac{B, \Gamma \xrightarrow{S} \Delta}{A \wedge B, \Gamma \xrightarrow{S} \Delta}$$

6) $$\frac{\Gamma \xrightarrow{S} \Delta, A \quad \Gamma \xrightarrow{S} \Delta, B}{\Gamma \xrightarrow{S} \Delta, A \wedge B}$$

7a) $$\frac{\Gamma \xrightarrow{S} \Delta, A}{\Gamma \xrightarrow{S} \Delta, A \vee B}$$

7b) $$\frac{\Gamma \xrightarrow{S} \Delta, B}{\Gamma \xrightarrow{S} \Delta, A \vee B}$$

8) $$\frac{A, \Gamma \xrightarrow{S} \Delta \quad B, \Gamma \xrightarrow{S} \Delta}{A \vee B, \Gamma \xrightarrow{S} \Delta}$$

9) $$\frac{\Gamma \xrightarrow{S} A_x[y], \Delta}{\Gamma \xrightarrow{S} \forall x A, \Delta}$$

where y is a variable not appearing free in Γ and $\forall x A$, and neither x nor y appear in S.

10) $$\frac{\Gamma, A_x[t] \xrightarrow{S} \Delta}{\Gamma, \forall x A \xrightarrow{S} \Delta}$$

where t is a term.

11) $$\frac{\Gamma, A_x[y] \xrightarrow{S} \Delta}{\Gamma, \exists x A \xrightarrow{S} \Delta}$$

where y is a variable not appearing free in Γ, $\exists x A$ and Δ, and y is a variable not appearing in S.

12) $$\frac{\Gamma \xrightarrow{S} A_x[t], \Delta}{\Gamma \xrightarrow{S} \exists x A, \Delta}$$

where x is a variable not appearing in S and t is a term.

13) $$\frac{\Gamma \longrightarrow \Delta_y[f(x_1,\dots,x_m)]}{\Gamma \xrightarrow{y:=f(x_1,\dots,x_m)} \Delta}$$

14) $$\frac{A,\ \Gamma \xrightarrow{S_1} \Delta \qquad \neg A,\ \Gamma \xrightarrow{S_2} \Delta}{\Gamma \xrightarrow{\underline{\text{if}}\ A\ \underline{\text{then}}\ S_1\ \underline{\text{else}}\ S_2} \Delta}$$

where A is of the form $P(x_1,\dots,x_n)$.

15) $$\frac{A,\ \Gamma \xrightarrow{S_1} \Gamma}{\Gamma \xrightarrow{\underline{\text{while}}\ A\ \underline{\text{do}}\ S_1} \Gamma,\ \neg A}$$

where A is of the form $P(x_1,\dots,x_n)$.

16) $$\frac{\Gamma_0 \xrightarrow{S_1} \Gamma_1 \quad \Gamma_1 \xrightarrow{S_2} \Gamma_2 \quad \dots \quad \Gamma_{n-1} \xrightarrow{S_n} \Gamma_n}{\Gamma_0 \xrightarrow{\underline{\text{begin}}\ S_1;\ S_2;\ \dots\ ;\ S_n\ \underline{\text{end}}} \Gamma_n}$$

The notion of provability in $\mathcal{S}$ is defined in the same way as LK.

Remark 2. Following two rules can be derived in $\mathcal{S}$.

i. $$\frac{\Gamma \xrightarrow{S} \Delta \qquad \Gamma \xrightarrow{S} \Delta'}{\Gamma \xrightarrow{S} \Delta, \Delta'}$$

ii. $$\frac{\Gamma \longrightarrow \Delta, \Pi \qquad \Pi, \Gamma' \longrightarrow \Delta'}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}$$

Theorem 3. If the sequent $\Gamma \longrightarrow A_1,\dots, A_m$ is provable in LK, then the sequent $\Gamma \longrightarrow A_1 \vee \dots \vee A_m$ is provable in $\mathcal{S}$. (When m = 0, i.e, $\Gamma \longrightarrow$ is provable in LK, $\Gamma \longrightarrow B$ is provable in $\mathcal{S}$ for any formula B.) Conversely, if $\Gamma \longrightarrow A_1,\dots, A_m$ is provable in $\mathcal{S}$ then $\Gamma \longrightarrow A_1 \wedge \dots \wedge A_m$ is provable in LK.

In order to deal with a program on a particular domain, e.g. a program on natural numbers, we need to define a theory on $\mathcal{S}$. A theory T on LK is defined as a system obtained by adding some sequents of the form $\longrightarrow A$ to LK as beginning sequents. In this case, such a formula A is called an axiom of T. A theory $T(\mathcal{S})$ on $\mathcal{S}$ is a system obtained from $\mathcal{S}$ by adding a beginning sequent $\longrightarrow A$ for every axiom of a theory T (on LK). Theorem 3 holds also for T and $T(\mathcal{S})$.

Next, we shall consider the systems in which countably infinite conjunctions and disjunctions are admitted. So, in these systems $\bigwedge_{j\in I} A_j$ and $\bigvee_{j\in I} A_j$ are formulas if I is at most countable.

Now, let LK* and $\mathcal{S}$* be the formal systems obtained from LK and $\mathcal{S}$, respectively, by changing the rules of inference concerned with conjunction and disjunction as follows. (For LK*, S is empty in the following.)

1a) $$\frac{A_i, \Gamma \xrightarrow{S} \Delta \quad \text{for some } i \in I}{\bigwedge_{j\in I} A_j, \Gamma \xrightarrow{S} \Delta}$$

1b) $$\frac{\Gamma \xrightarrow{S} A_i, \Delta \quad \text{for any } i \in I}{\Gamma \xrightarrow{S} \bigwedge_{j\in I} A_j, \Delta}$$

2a) $$\frac{A_i, \Gamma \xrightarrow{S} \Delta \quad \text{for any } i \in I}{\bigvee_{j\in I} A_j, \Gamma \xrightarrow{S} \Delta}$$

2b) $$\frac{\Gamma \xrightarrow{S} A_i, \Delta \quad \text{for some } i \in I}{\Gamma \xrightarrow{S} \bigvee_{j\in I} A_j, \Delta}$$

We can prove also that Theorem 3 holds for LK* and $\mathcal{S}$*.

2. Interpretation of $\mathcal{S}$ in LK*

In this section, we shall define an interpretation Φ of each sequent of $\mathcal{S}$. For each sequent $\Gamma \xrightarrow{S} \Delta$ of $\mathcal{S}$, a sequent $\Phi(\Gamma \xrightarrow{S} \Delta)$ of LK* is defined so that $\Phi(\Gamma \xrightarrow{S} \Delta)$ is provable in LK* if $\Gamma \xrightarrow{S} \Delta$ is provable in $\mathcal{S}$. Thus, we can say that every sequent provable in $\mathcal{S}$ is 'true'. As shown in the following, our interpretation has a close relation with the verification condition of the inductive assertion method.

Let S be a statement or empty. We define a formula $\varphi_S(A)$ of LK* for each formula A of $\mathcal{S}$ as follows.

1) $\varphi_S(A) \equiv A$ if S is empty.

2) $\varphi_S(A) \equiv A_y[f(x_1,\ldots,x_m)]$ if S is $y:=f(x_1,\ldots,x_m)$.

3) $\varphi_S(A) \equiv (P(x_1,\ldots,x_n) \wedge \varphi_{S_1}(A)) \vee (\neg P(x_1,\ldots,x_n) \wedge \varphi_{S_2}(A))$ if S is <u>if</u> $P(x_1,\ldots,x_n)$ <u>then</u> S_1 <u>else</u> S_2.

4) $\varphi_S(A) \equiv \bigwedge_{n=0}^{\infty} \sigma_n(A)$ if S is <u>while</u> $P(x_1,\ldots,x_n)$ <u>do</u> S_1, where $\sigma_n(A)$ is defined by

$$\begin{cases} \sigma_0(A) \equiv \neg P(x_1,\ldots,x_n) \supset A \\ \sigma_{n+1}(A) \equiv P(x_1,\ldots,x_n) \supset \varphi_{S_1}(\sigma_n(A)). \end{cases}$$

5) $\varphi_S(A) \equiv \varphi_{S_1}(\varphi_{S_2}(\ \ldots\ (\varphi_{S_n}(A))\ \ldots\ ))$ if S is <u>begin</u> S_1; S_2; ... ; S_n <u>end</u>.

Next, define Φ by

$$\Phi(\ \Gamma \xrightarrow{S} A_1,\ldots, A_m\ ) \equiv \Gamma \longrightarrow \bigwedge_{i=1}^{m} \varphi_S(A_i).$$

We can see that $\Phi(\ \Gamma \xrightarrow{S} \Delta\ )$ is the verfication condition for the statement S.

Theorem 4. If a sequent $\Gamma \xrightarrow{S} \Delta$ is provable in $\mathcal{S}$, then $\Phi(\ \Gamma \xrightarrow{S} \Delta\ )$ is provable in LK*.

We don't know whether the converse of Theorem 4 holds. We can only show that when the statement S contains no <u>while</u> ... <u>do</u> ...-statements the converse holds, by using the cut-elimination theorem of LK*. On the other hand, we have the following theorem.

Theorem 5. $\Phi(\ \Gamma \xrightarrow{S} \Delta\ )$ is provable in LK* iff $\Gamma \xrightarrow{S} \Delta$ is provable in $\mathcal{S}$*, where Γ and Δ are sets of formulas of LK*.

The above theorem means that the system $\mathcal{S}$* is 'complete'.

References

[1] G. Gentzen, Untersuchungen über das Logische Schliessen, Math. Zeitschrift <u>39</u> (1934/35) 176 - 210, 405 - 431.

[2] C. A. R. Hoare, An axiomatic basis of computer programming, CACM <u>12</u> (1969) 576 - 580, 583.

[3] Z. Manna, Mathematical theory of computation, 1974, McGraw-Hill.

[4] H. Ono, プログラムの基礎理論 (Introduction to mathematical theory of computation, in Japanese), Seminar note on mathematics, No.2, Tsuda College, 1974.

(Revised February 1975)